

```

#define NUMESTADOS 15
#define NUMCOLS 13
#define TAMLEX 32+1
#define TAMNOM 20+1
/*****Declaraciones Globales*****/
FILE * in;
typedef enum {
    INICIO, FIN, LEER, ESCRIBIR, ID, CONSTANTE, PARENIZQUIERDO, PARENDERECHO, PUNTOYCOMA,
    COMA, ASIGNACION, SUMA, RESTA, FDT, ERRORLEXICO
} TOKEN;
/*****Scanner*****/
TOKEN scanner()
{
    int tabla[NUMESTADOS][NUMCOLS] =
        L D + - ( ) , ; : = EOF ' ' OTRO
0      { { 1, 3, 5, 6, 7, 8, 9, 10, 11, 14, 13, 0, 14 },
1      { 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 },
2 ID   { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
3      { 4, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4 },
4 CTE  { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
5 +    { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
6 -    { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
7 (    { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
8 )    { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
9 ,    { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
10 ;   { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
11     { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 12, 14, 14 },
12 ASIG{ 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
13 fdt { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 },
14 Err { 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14 } };
    int car;
    int col;
    int estado = 0;
    int i = 0;
    do {
        car = fgetc(in);
        col = columna(car);
        estado = tabla[estado][col];
        if ( col != 11 ) { //si es espacio no lo agrega al buffer
            buffer[i] = car;
            i++;
        }
    }
    while ( !estadoFinal(estado) && !(estado == 14) );
    buffer[i] = '\0'; //complete la cadena
    switch ( estado )
    {

```

```

case 2 : if ( col != 11 ){           //si el carácter espureo no es blanco...
    ungetc(car, in);               // lo retorna al flujo
    buffer[i-1] = '\0';
}
return ID;
case 4 : if ( col != 11 ) {
    ungetc(car, in);
    buffer[i-1] = '\0';
}
return CONSTANTE;
case 5 : return SUMA;
case 6 : return RESTA;
case 7 : return PARENIZQUIERDO;
case 8 : return PARENDERECHO;
case 9 : return COMA;
case 10 : return PUNTOYCOMA;
case 12 : return ASIGNACION;
case 13 : return FDT;
case 14 : return ERRORLEXICO;
}
return 0;
}

```

```

int estadoFinal(int e){
    if ( e == 0 || e == 1 || e == 3 || e == 11 || e == 14 ) return 0;
    return 1;
}

```

```
int columna(int c){  
    if ( isalpha(c) ) return 0;  
    if ( isdigit(c) ) return 1;  
    if ( c == '+' ) return 2;  
    if ( c == '-' ) return 3;  
    if ( c == '(' ) return 4;  
    if ( c == ')' ) return 5;  
    if ( c == ';' ) return 6;  
    if ( c == ':' ) return 7;  
    if ( c == '.' ) return 8;  
    if ( c == '=' ) return 9;  
    if ( c == EOF ) return 10;  
    if ( isspace(c) ) return 11;  
    return 12;  
}
```

/*****Fin Scanner*****/